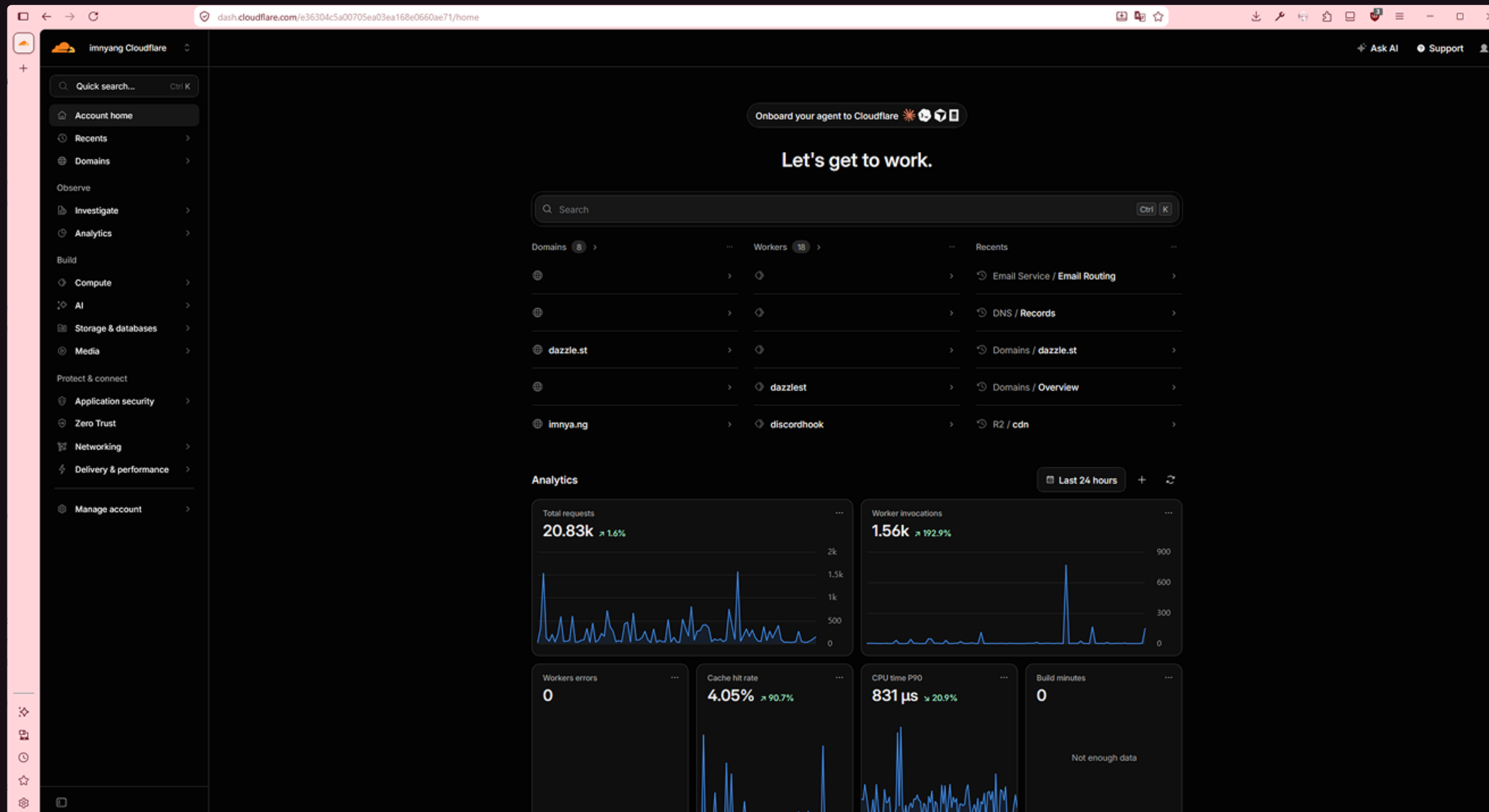


OAuth 인증 및 인가 취약점 탐지 자동화 도구 개발

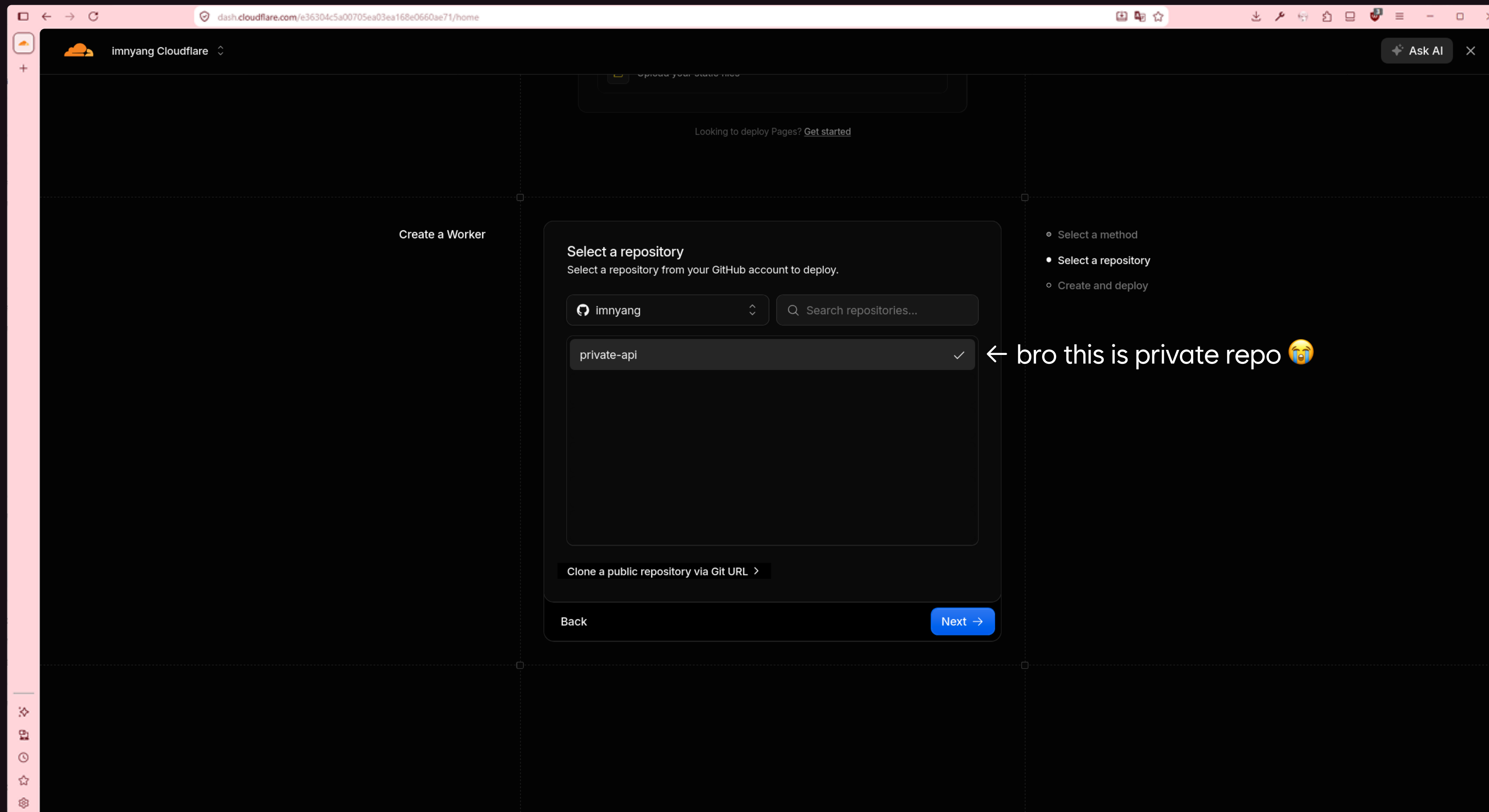
| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

이렇게 Cloudflare라는 코드만 주면 웹 서비스 배포해주는 서비스가 있어요. (더 많은 일을 하지만 생략)



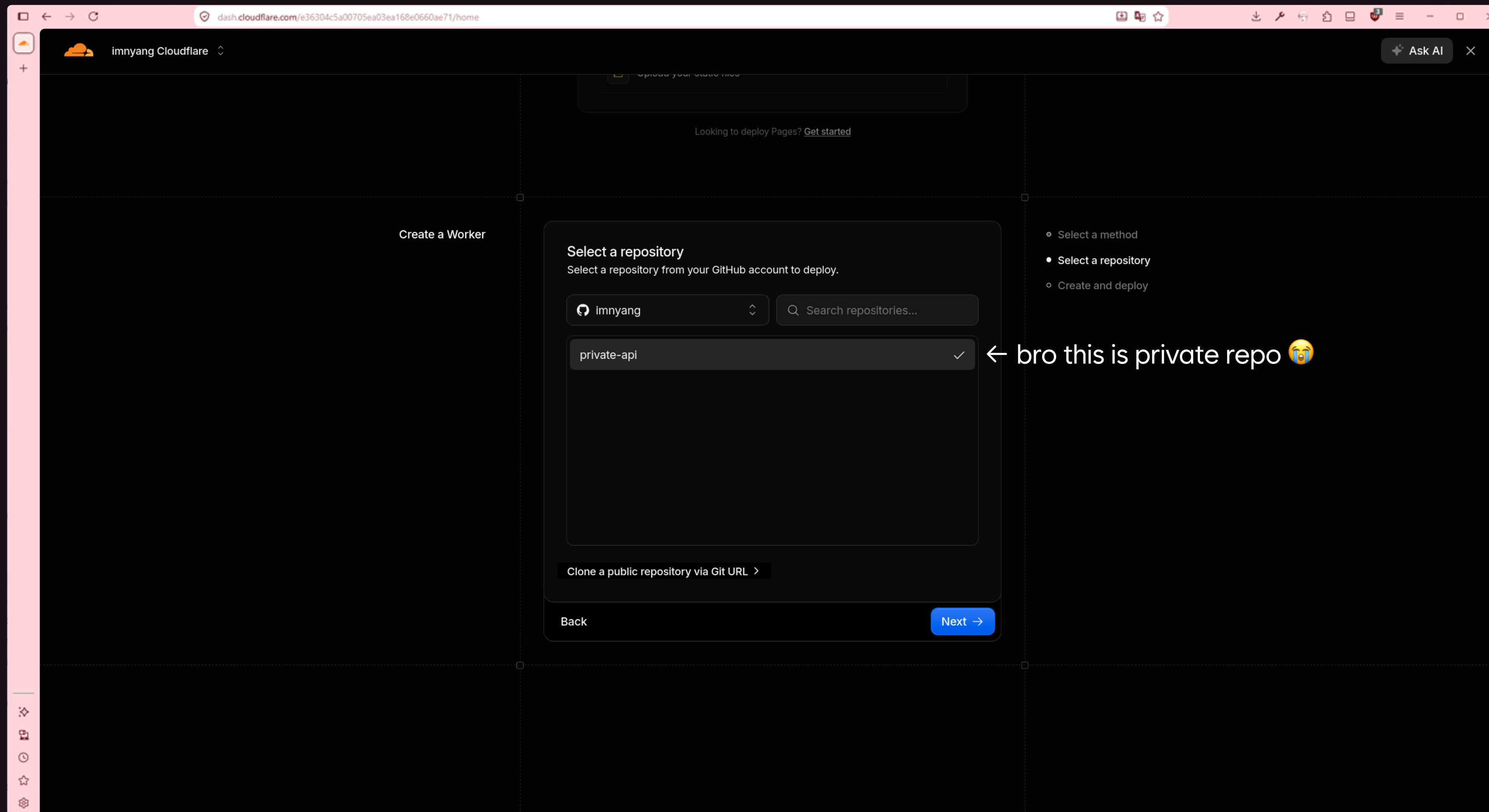
| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

아니 근데 잠깐 제 GitHub 레피지토리는 어떻게 접근하고 있는거죠?



| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

어떻게 서비스들은 다른 서비스들의 데이터를 가져올 수 있는 걸까요?



| 그럼 나는 내 계정에 어떻게 접근할까?

우리가 계정을 접근하는 방법은 무엇이 있을까요?

| 그럼 나는 내 계정에 어떻게 접근할까?

우리는 주로 사용자 이름, 비밀번호를 입력해서 토큰이나 세션을 받는 방식으로 접근해요.

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

그럼 서비스한테 다른 서비스의 내 비밀번호를 주면 되겠구나!

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

~~그럼 서비스한테 다른 서비스의 내 비밀번호를 주면 되겠구나!~~

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

왜 내 비밀번호를 그냥 주는 방식을 쓰지 않을까?

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

일단 권한이 너무 광범위적이고 관리가 어렵다

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

그럼 어떤 다른 방법이 있을까?

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

OAuth
API Key

Personal Access Token

...

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

OAuth

API Key

Personal Access Token

...

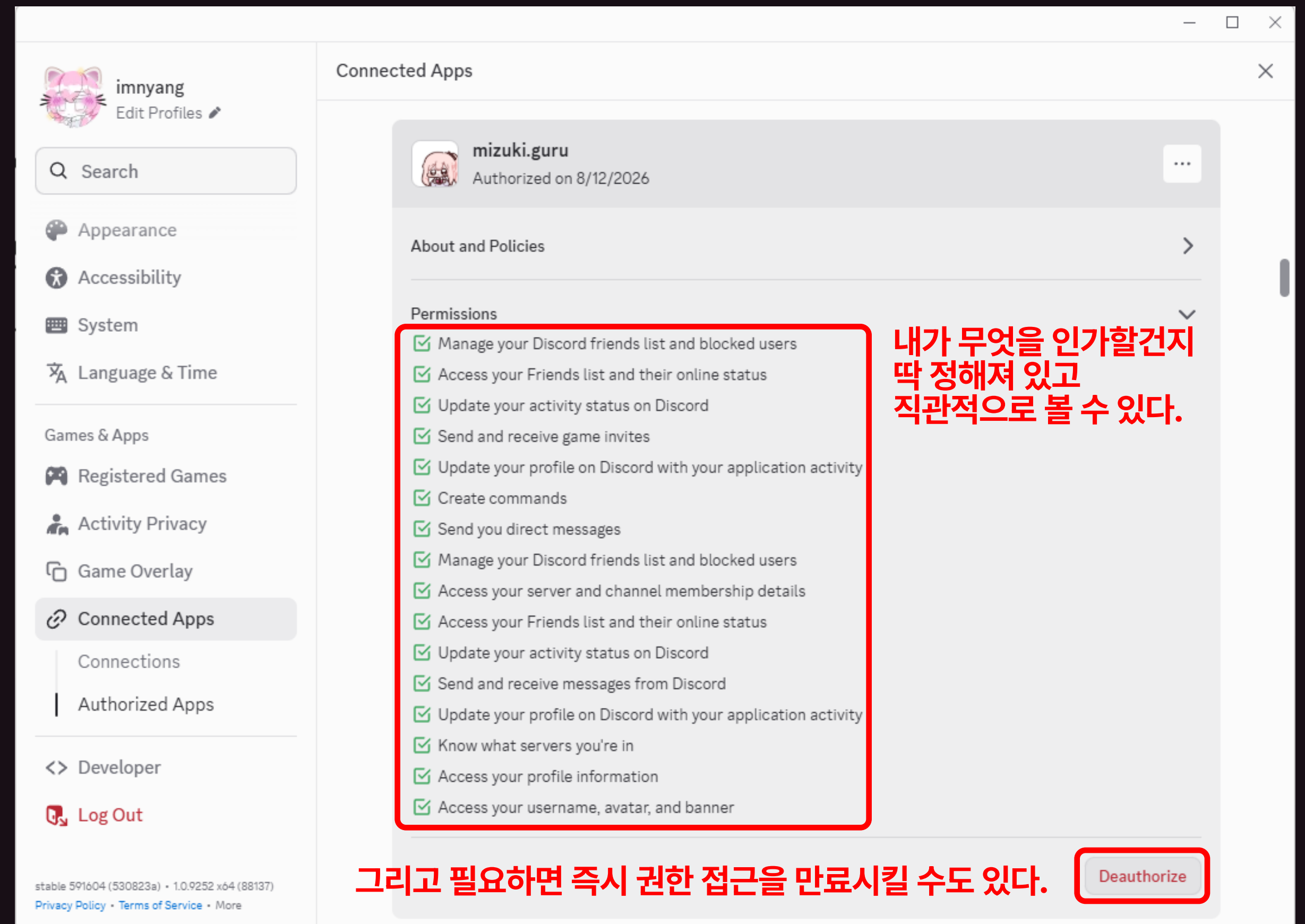
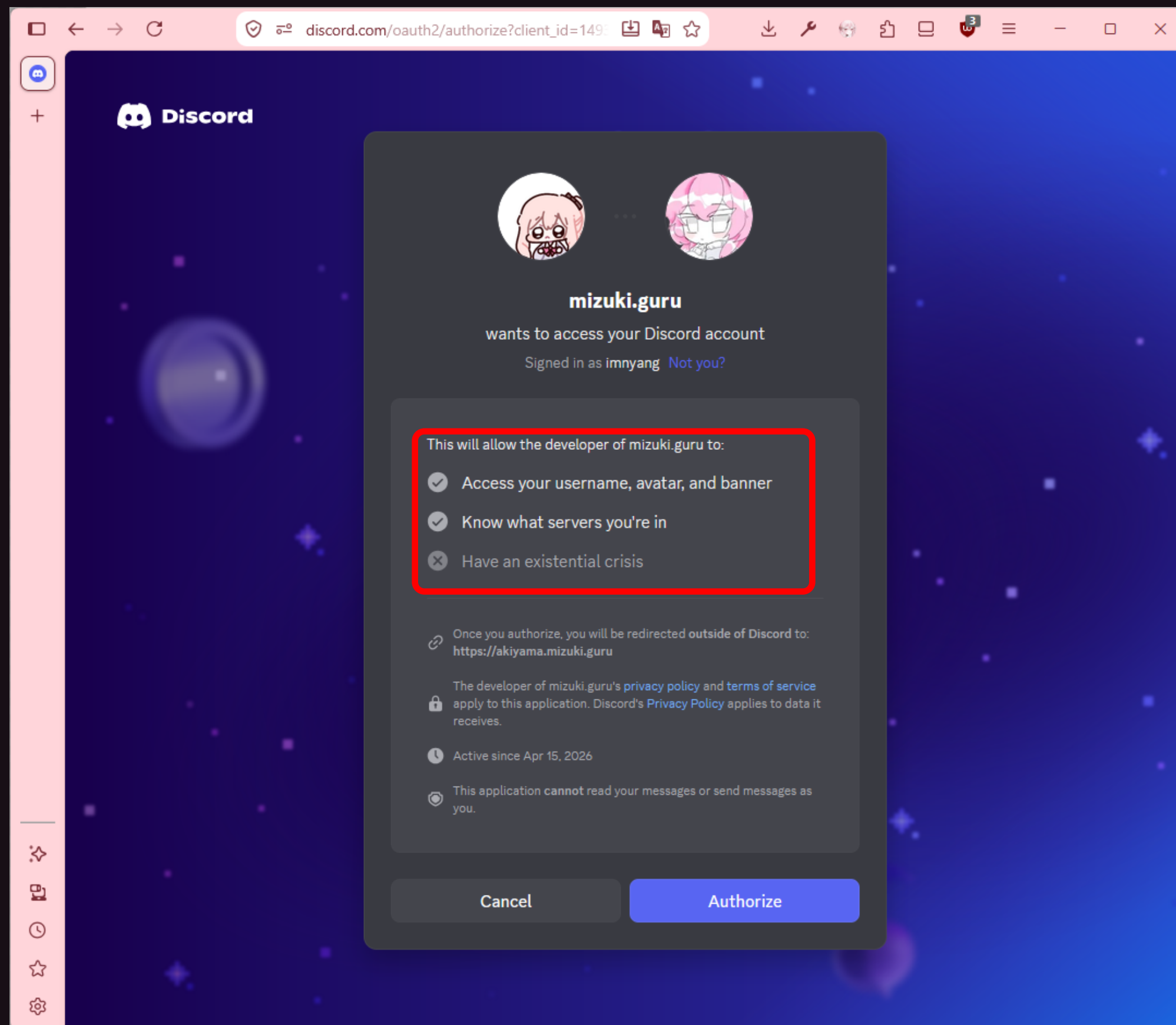
| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

근데 왜 OAuth일까?

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

근데 왜 OAuth일까?

권한 관리가 다른 것에 비해 압도적으로 쉬움

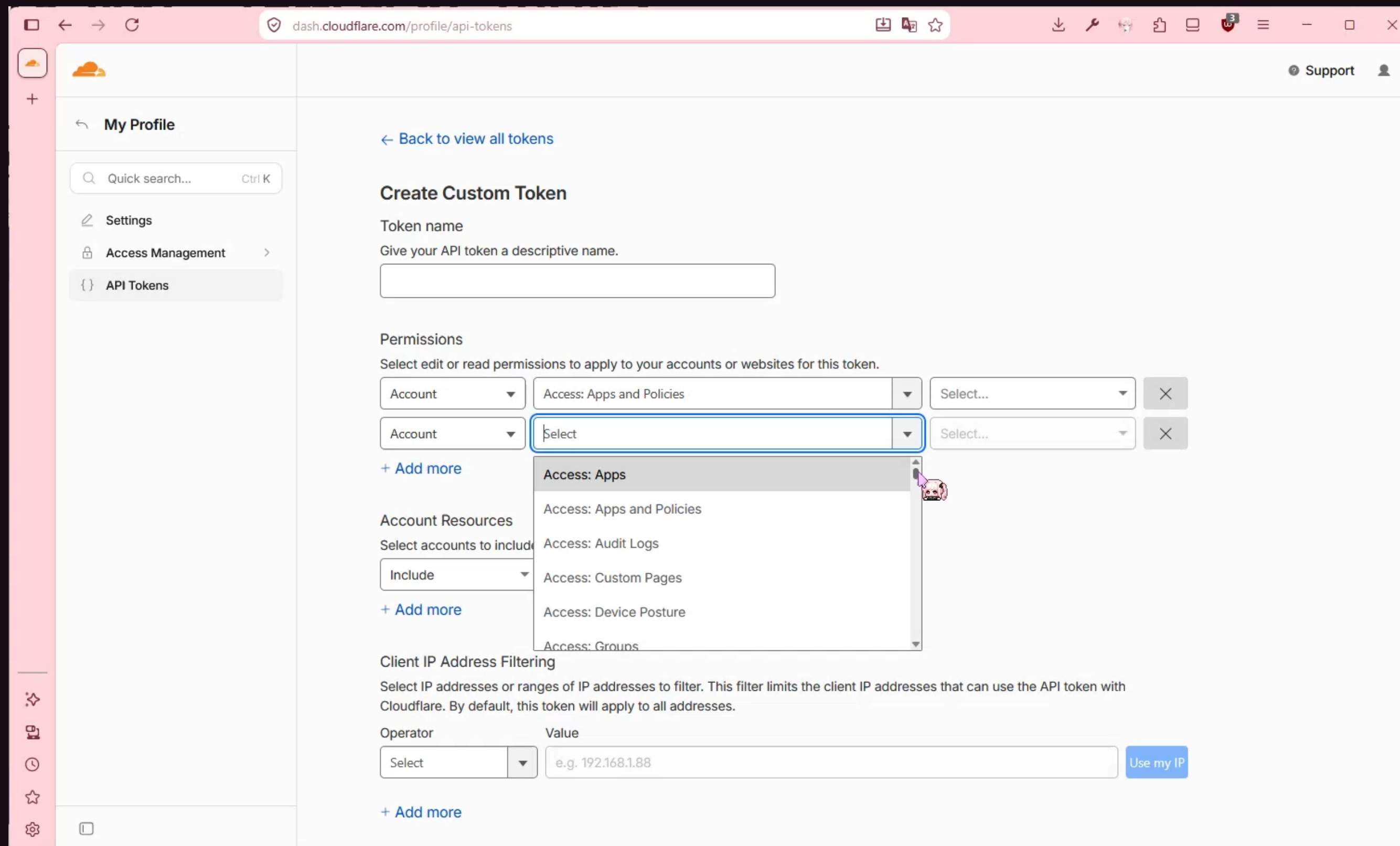


그리고 필요하면 즉시 권한 접근을 만료시킬 수도 있다.

| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

근데 왜 OAuth일까?

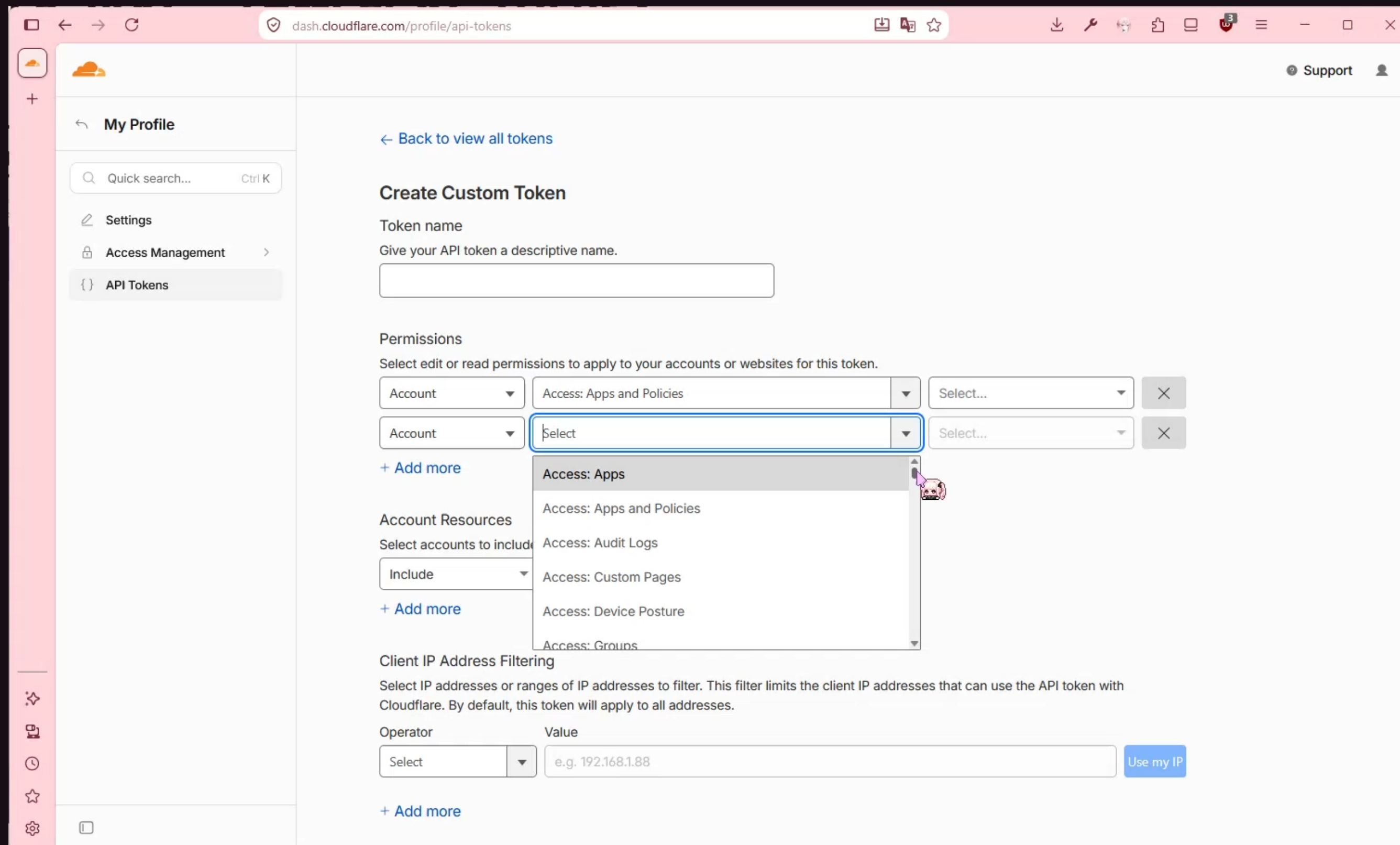
권한 관리가 다른 것에 비해 압도적으로 쉬움
만약 API Key를 쓴다면?



| 서비스들은 내 다른 서비스의 계정을 어떻게 접근할까?

근데 왜 OAuth일까?

권한 관리가 다른 것에 비해 압도적으로 쉬움
만약 API Key를 쓴다면?



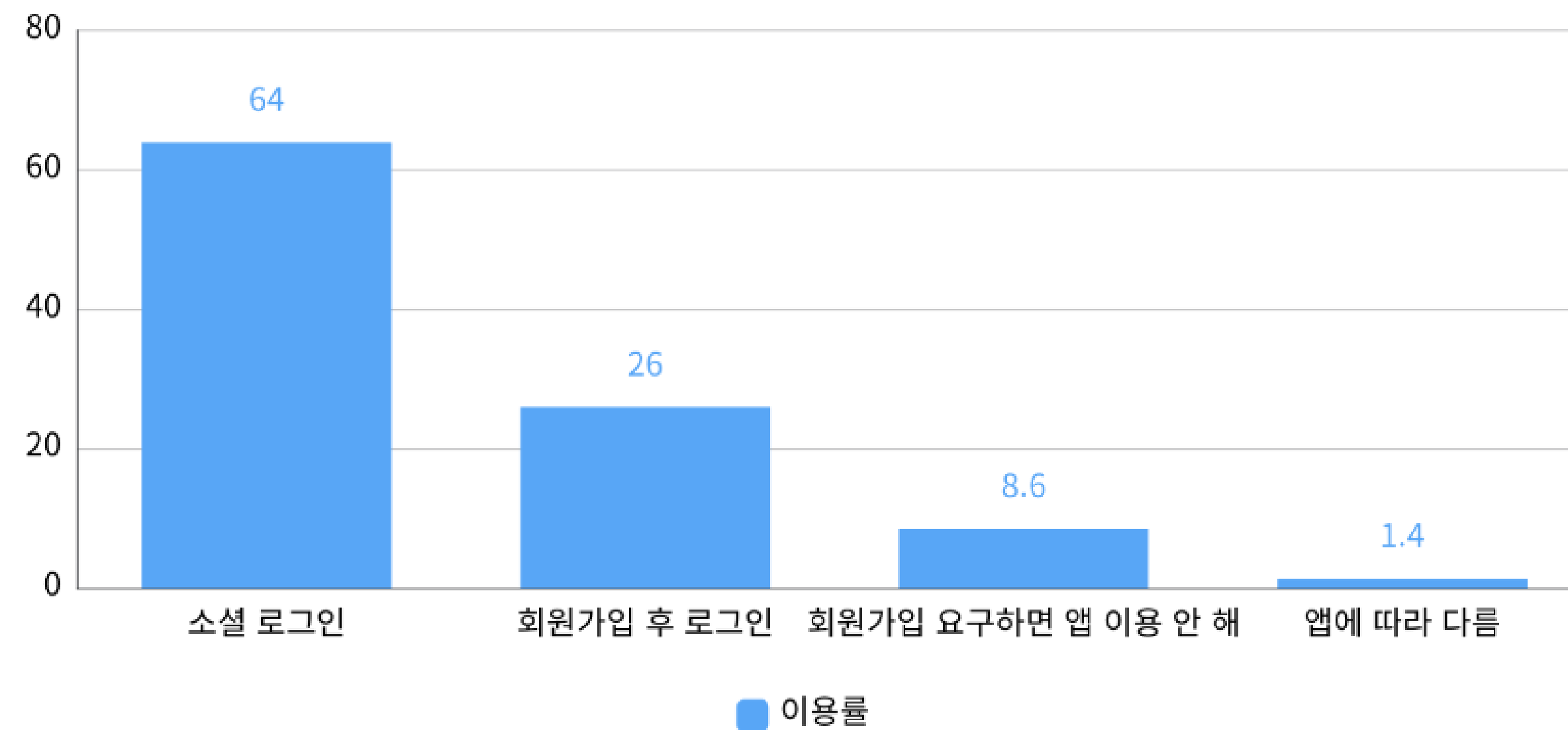
보통 서비스들의 권한이 정말로 많이 있다...
사용자가 권한을 혼동해서
필요 이상으로 권한을 줄 수도 있어요.

그리고 개발자한테 편할 수도 있어도
일반 사용자한테 좀...

| OAuth란?

여러분들은 OAuth를 써보신적이 있으신가요?

| OAuth란?



한국 소비자 연맹(2020), 소셜 로그인 인식도 조사,
[arxiv abs/2111.03573](https://arxiv.org/abs/2111.03573) Security and Privacy Perceptions of Third-Party Application Access for Google Accounts (Extended Version)

| OAuth란?

Sign up

Email

Auth0 may contact me with marketing communications
opt-out details in [Privacy Policy](#).

☐ I agree

By continuing, you agree to the [Self Service PSS](#) and
[Privacy Policy](#).

Continue

OR



Continue with GitHub



Continue with Google



Continue with Microsoft

| OAuth란?

Sign up

Email

Auth0 may contact me with marketing communications
opt-out details in [Privacy Policy](#).

☐ I agree

By continuing, you agree to the [Self Service PSS](#) and
[Privacy Policy](#).

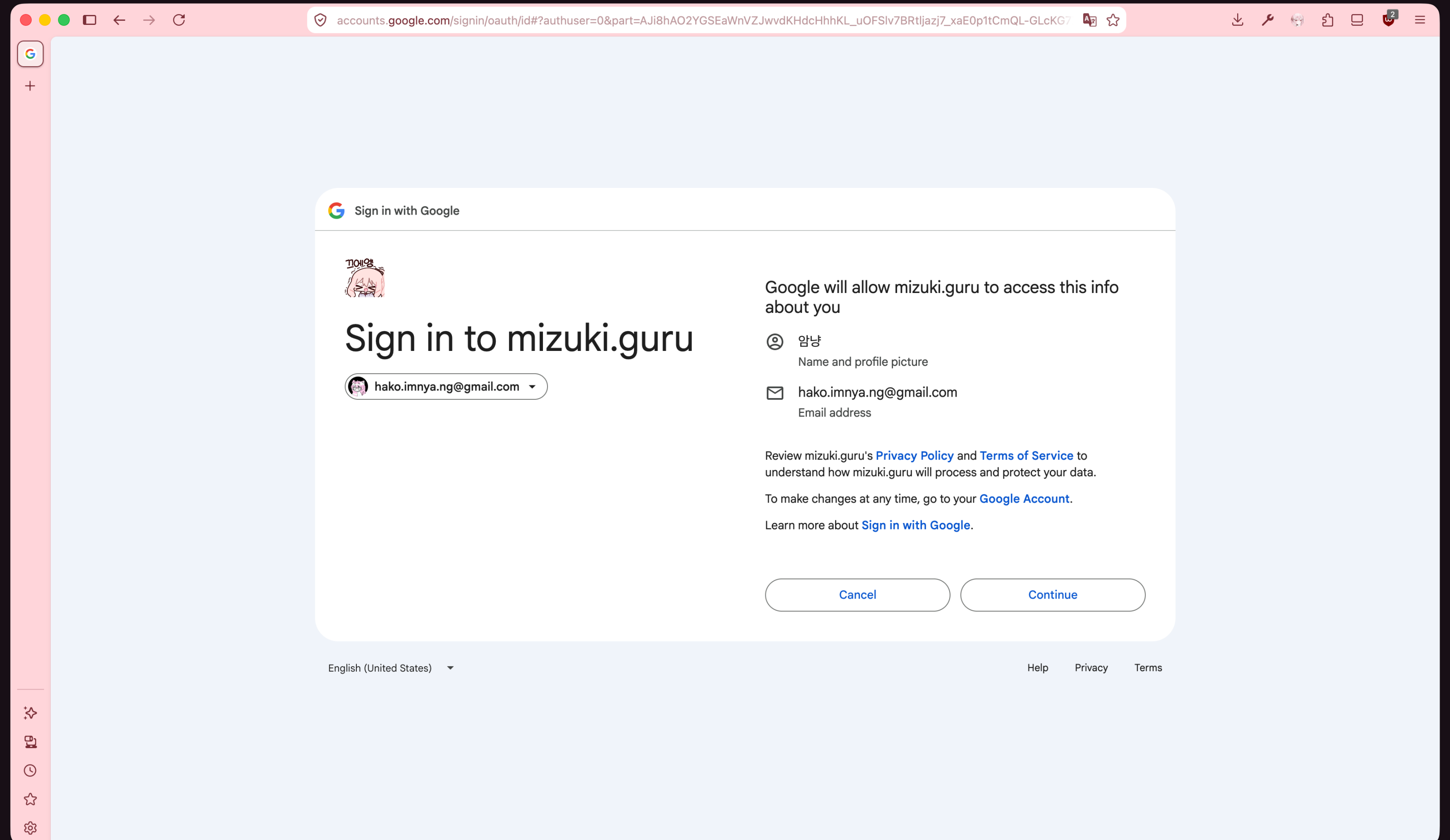
Continue

OR

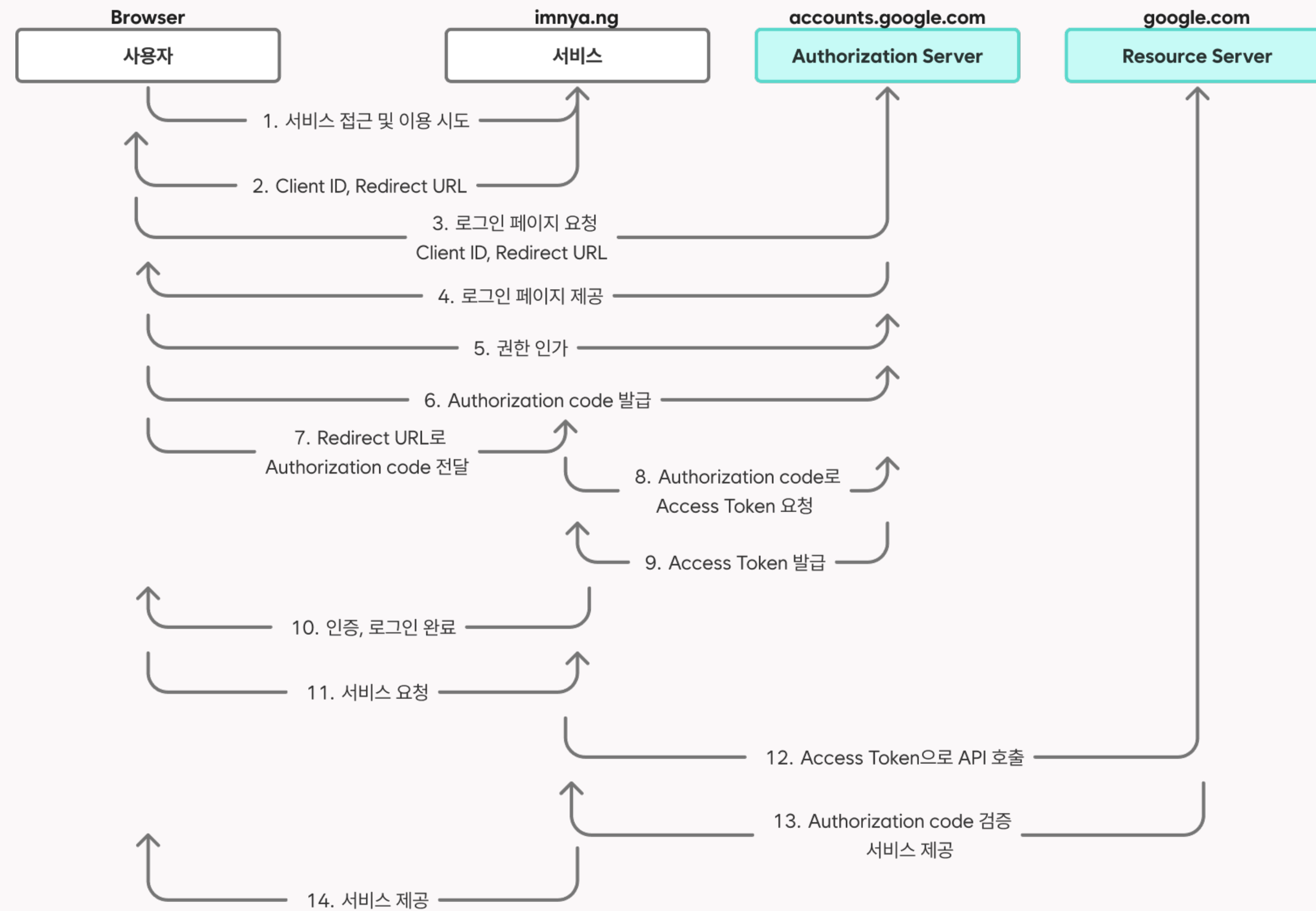
 Continue with GitHub

 Continue with Google

 Continue with Microsoft

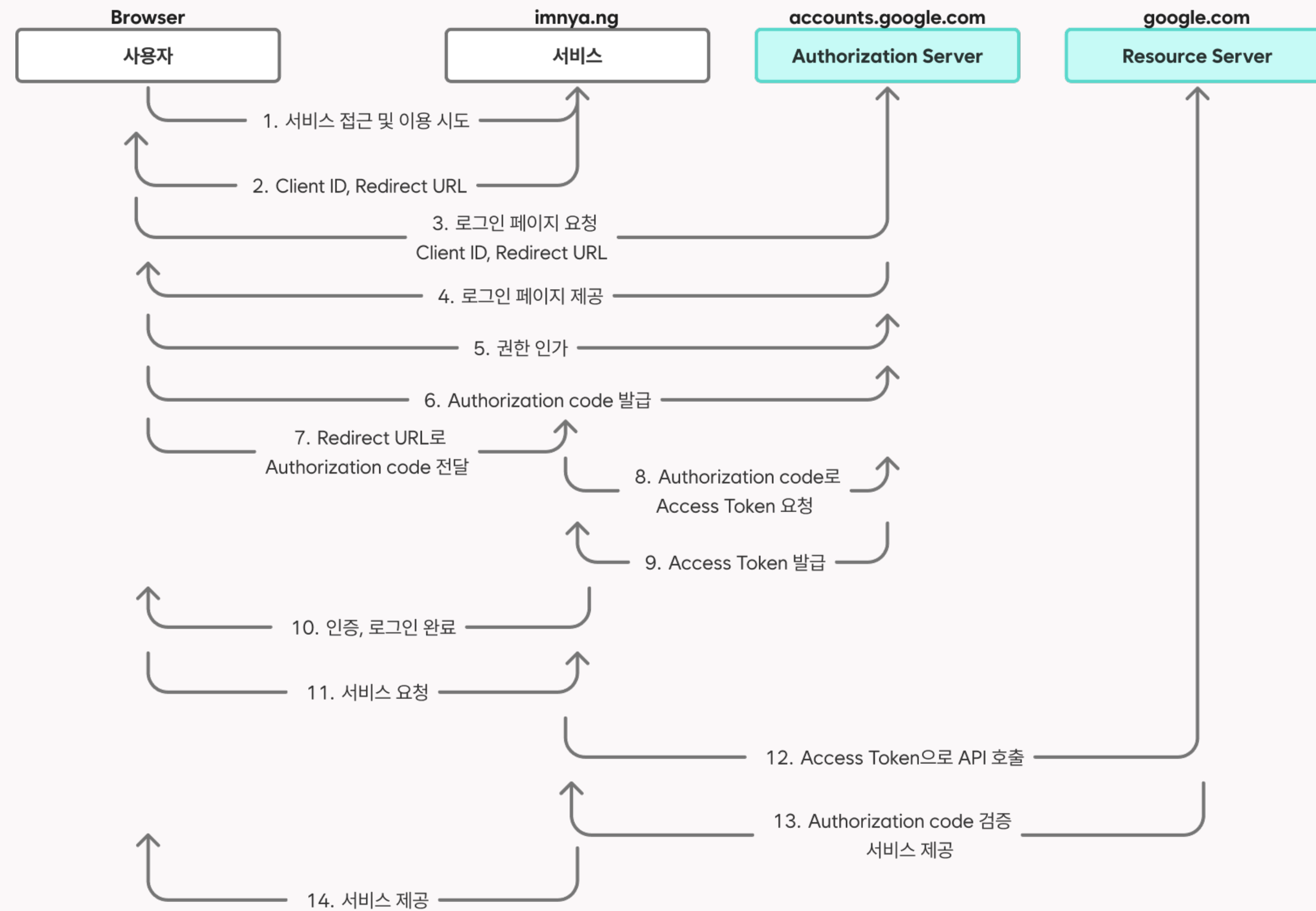


| OAuth란?



| OAuth란?

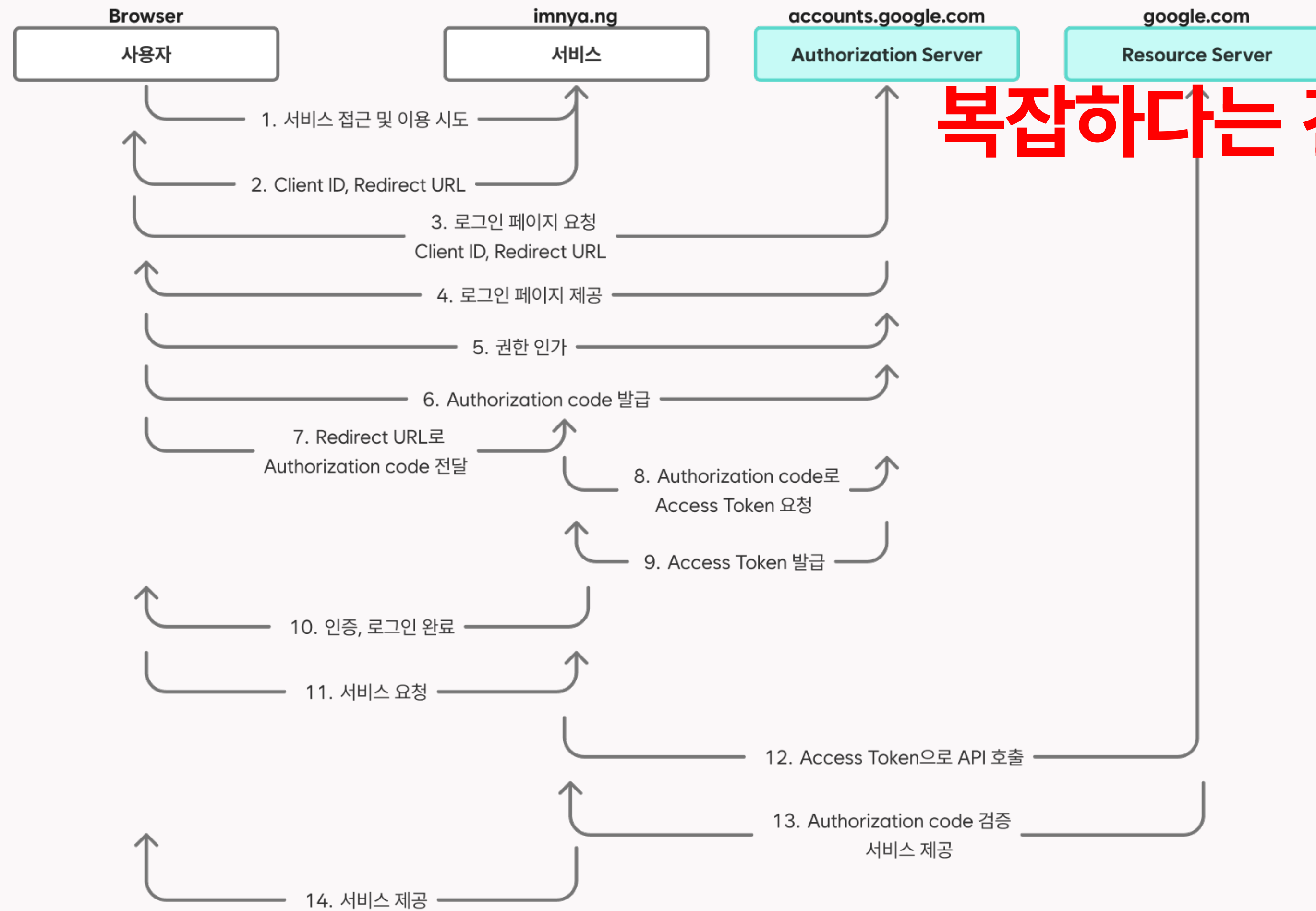
굉장히 복잡함



| OAuth란?

굉장히 복잡함

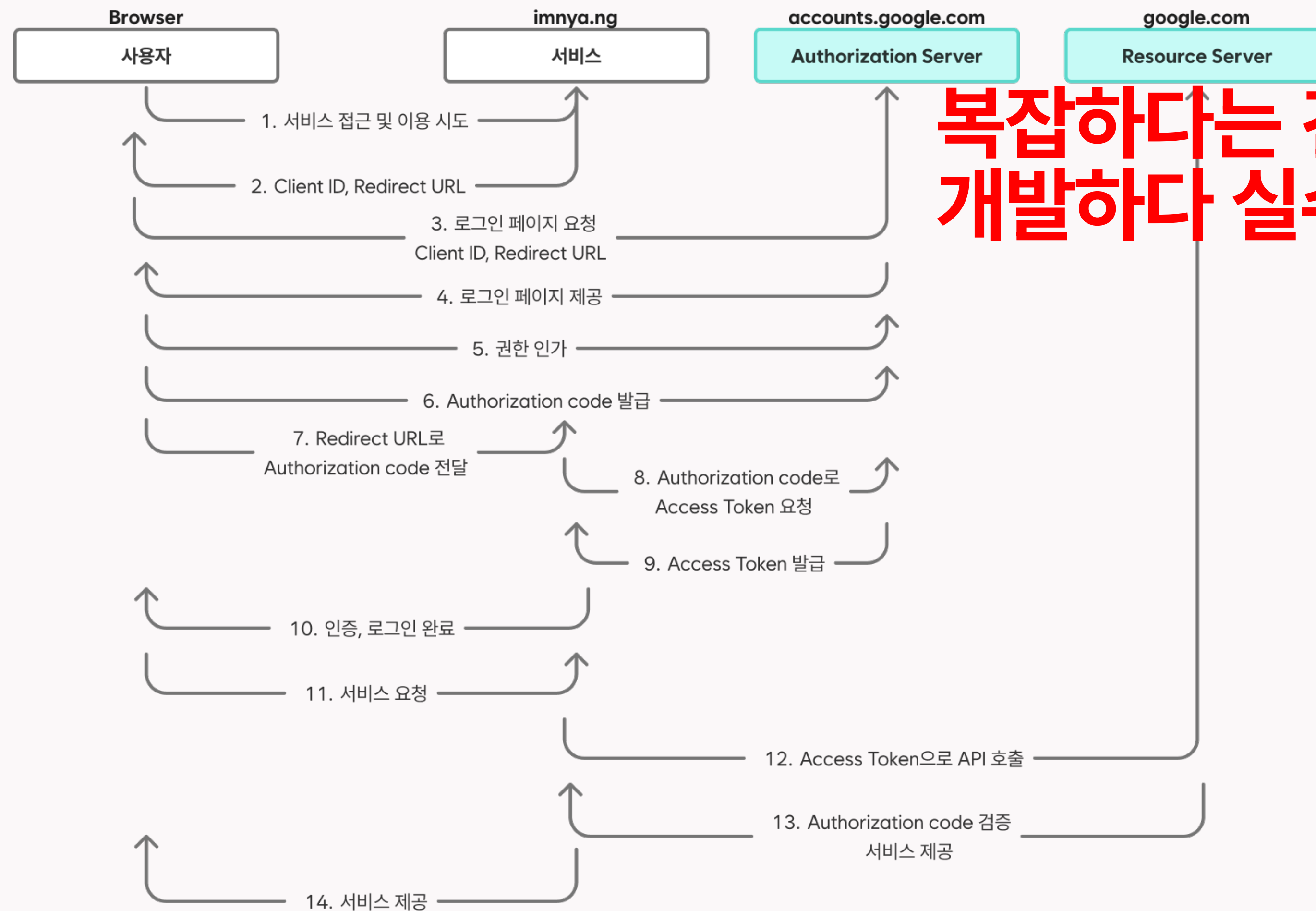
복잡하다는 건 = 개발하다 실수가 많이 난다



| OAuth란?

굉장히 복잡함

복잡하다는 건 = 개발하다 실수가 많이 난다
개발하다 실수가 난다 = 취약점 생성



| OAuth란?

VULNERABLE

```
if (uri.startsWith(  
    "https://app.imnyang.dev"  
) allow();
```

SAFE

```
registered_uri ===  
    received_uri  
// exact string match
```

| OAuth란?

VULNERABLE

```
if (uri.startsWith(  
    "https://app.imnyang.dev"  
)) allow();
```

`https://app.imnyang.dev.hotline.tld/callback`

SAFE

```
registered_uri ===  
    received_uri  
// exact string match
```

`https://app.imnyang.dev/callback`

| 자동화 도구 제작

그럼 이런 취약점을 자동으로 탐지해보면 어떨까?

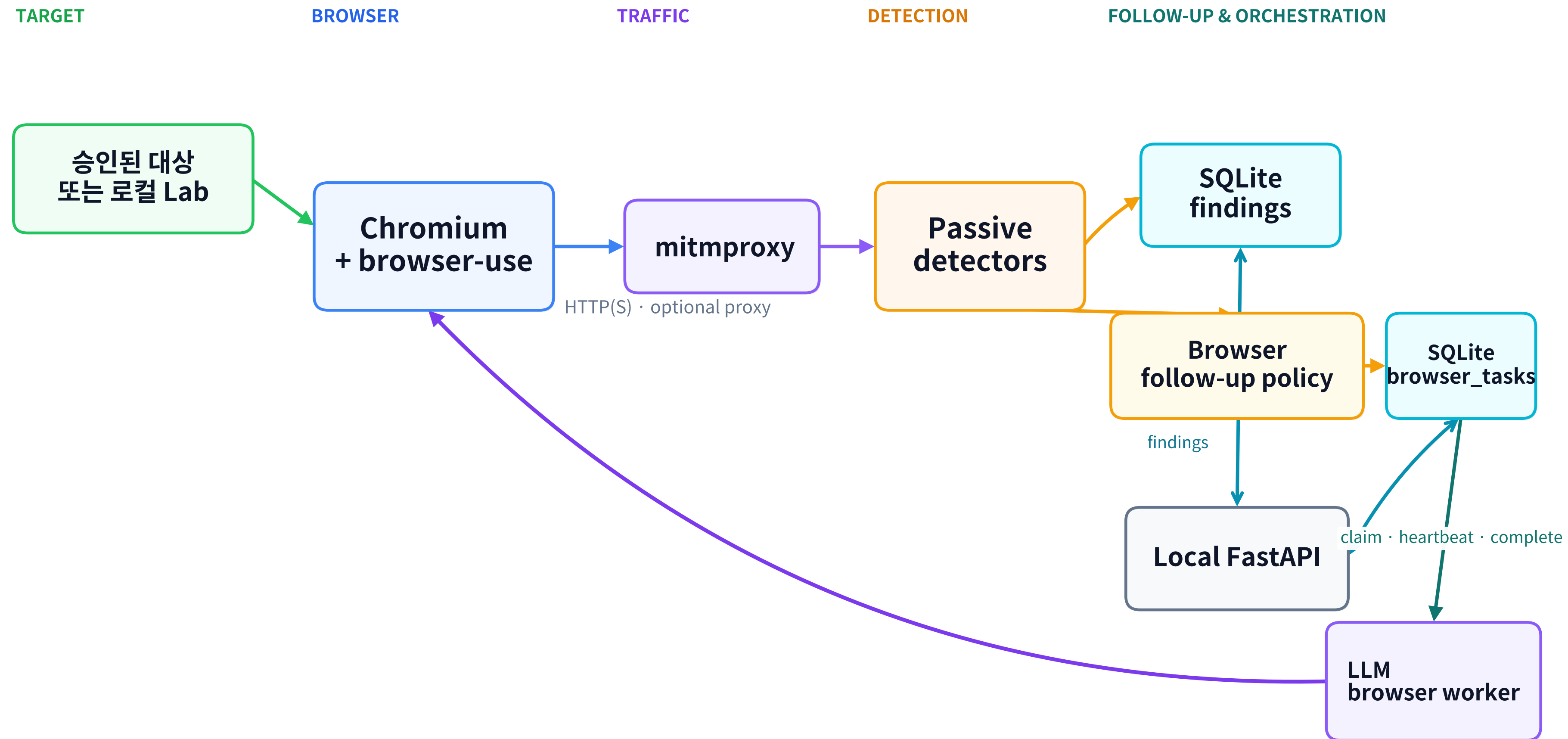
| 자동화 도구 제작

MITMproxy
Browser Use

| 자동화 도구 제작

MITMproxy
Browser Use

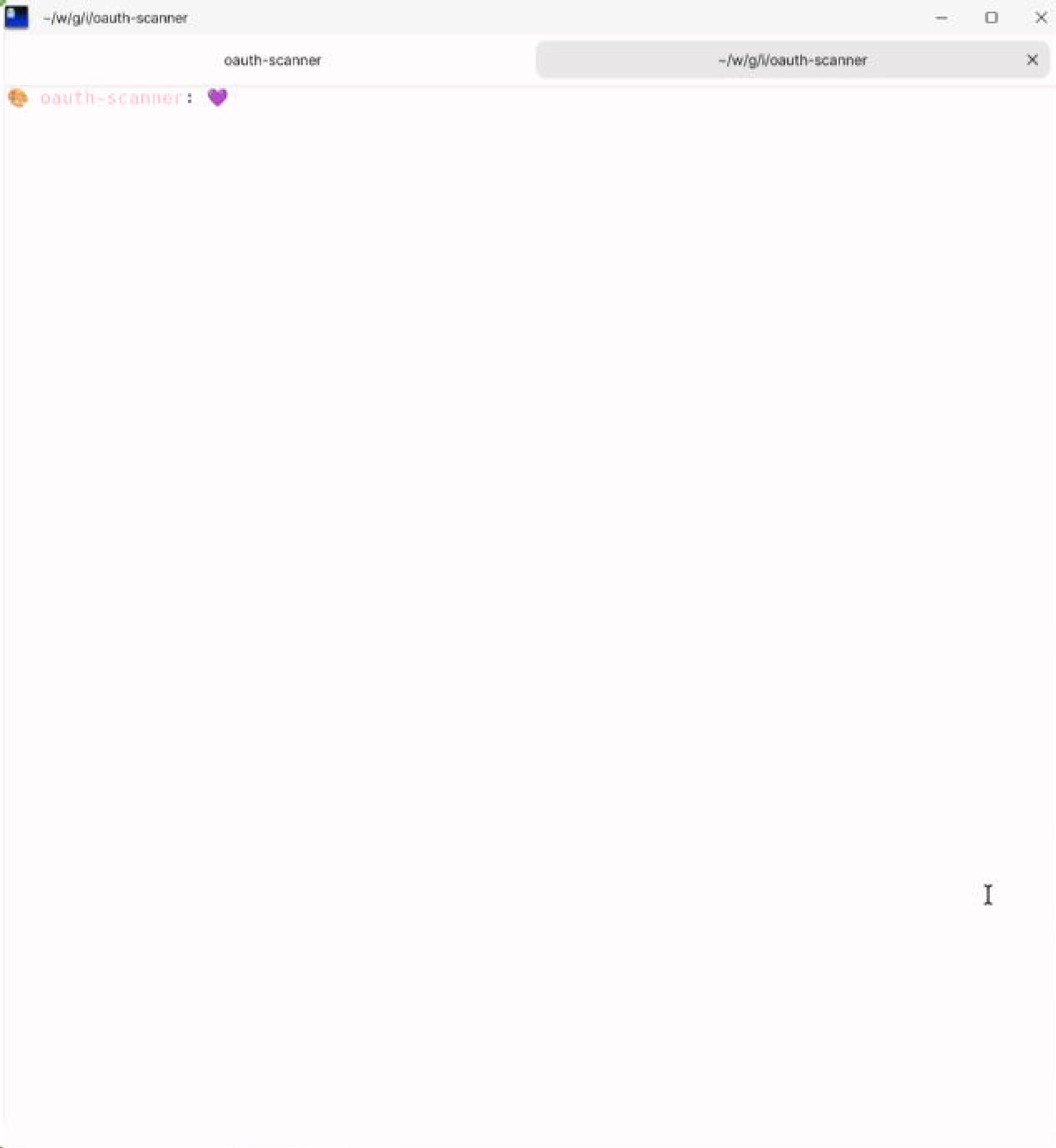
| 자동화 도구 제작



Passive detection → findings → allowlisted browser follow-up

| 자동화 도구 제작

규칙 ID	관찰하는 증거	관련 분류
oauth.authorization-over-http	authorization request가 HTTPS가 아님	전송 보안
oauth.secret-over-http	token exchange/client credential이 HTTP로 전송됨	전송 보안
oauth.bearer-over-http	Bearer Authorization header가 HTTP로 전송됨	bearer token
oauth.secret-in-query	token, verifier, secret이 URL query에 있음	URL·로그 유출
oauth.code-in-query	authorization code가 URL에 있어 leak surface가 생김	code leakage (낮은 신뢰도)
oauth.secret-in-referer	실제 Referer에 code/token이 보임	실제 credential 유출
oauth.password-grant	grant_type=password	ROPC
oauth.duplicate-authorization-parameter	보안 중요 authorization parameter 중복	parameter pollution
oauth.state-missing / oauth.state-not-returned	state 없음 또는 callback에서 원값을 반환하지 않음	CSRF/login CSRF
oidc.nonce-missing	OIDC 요청에 nonce 없음	ID token replay/code injection 방어
oauth.pkce-missing / oauth.pkce-weak	code flow PKCE 없음 또는 S256 이외 방식	code interception/downgrade
oauth.implicit-flow	token response type(혼합 flow 포함)	front-channel access token
oauth.token-in-redirect / oauth.code-in-fragment	redirect query/fragment의 token 또는 fragment code	browser credential leak
oauth.redirect-uri-mismatch	authorization response의 callback 대상이 요청한 redirect URI와 다름	redirect validation
oauth.redirect-uri-fragment	redirect URI에 fragment 포함	잘못된 callback URI
oauth.duplicate-authorization-response-parameter	callback response parameter 중복	response injection/parameter pollution
oauth.credential-forwarded-in-redirect	받은 code/token을 다음 redirect가 그대로 전달	open redirect/credential forwarding
oauth.insecure-redirect-uri	native loopback 외 HTTP redirect URI	code/token interception
oauth.token-response-cacheable	token response의 no-store 또는 no-cache 누락	cache leakage
oauth.authorization-ui-frameable	authorization HTML에 CSP frame-ancestors 와 XFO가 모두 없음	clickjacking
oauth.wildcard-scope	* 또는 all scope	최소 권한 위반 신호



감사합니다